

COMPOSANT À ARCHITECTURE RECONFIGURABLE DYNAMIQUEMENT**DESCRIPTION****5 DOMAINE TECHNIQUE**

L'invention relève du domaine des composants reconfigurables dynamiques dédiés au traitements de données, notamment pour l'exécution des instructions correspondant aux diverses tâches d'une
10 application.

ÉTAT DE LA TECHNIQUE ANTÉRIEURE

Les composants reconfigurables de manière
15 dynamique ont classiquement pour finalité une utilisation optimisée de leurs ressources matérielles lors de traitements de données effectués pour réaliser une application. Le plus souvent il s'agit de reconfiguration visant à obtenir un taux d'utilisation
20 d'un composant le plus élevé possible. D'autres fois il s'agit d'utiliser la reconfiguration pour viser la robustesse d'un composant (c'est-à-dire de permettre au composant de fonctionner même en cas de défaillance d'une partie de ses ressources matérielles), ou bien il
25 s'agit de viser un très haut débit de données traitées par le composant ou bien encore une consommation d'énergie réduite lors de l'utilisation du composant. La reconfiguration peut aussi être utilisée pour optimiser le traitement de données en parallèle sur des
30 processeurs élémentaires du composant.

Il existe différentes manières de réaliser une application : avec un circuit intégré à application spécifique (un ASIC, 'Application Specific Integrated Circuit') dédié, avec un processeur ou avec un réseau
5 prédiffusé programmable par l'utilisateur (un FPGA, 'Field Programmable Gate Array').

L'ASIC dédié offre une réalisation spatiale optimum puisque les opérateurs sont directement câblés sur le silicium ; cependant ce type de composant étant
10 dédié à une application donnée, il ne permet pas de passer (ou alors de façon très dégradée) à un autre type d'application.

L'utilisation d'un processeur permet une implémentation temporelle d'une application : celle-ci
15 est traduite en séquences d'instructions réalisables par le processeur, avec réutilisation de son unité de traitement à chaque cycle. Suivant le jeu d'instructions du processeur, cette technique offre une grande souplesse au niveau des applications qui peuvent
20 être traitées et de plus l'architecture est très compacte. L'inconvénient principal du processeur concerne la performance qui est très loin de celle des ASICs dédiés.

Les composants programmables (on dit aussi
25 reconfigurables) que sont les FPGA présentent une alternative entre le processeur et l'ASIC : une application est implantée sur une matrice de cellules de traitement pré-caractérisées (grains fins du composant) avec de nombreuses interconnexions.
30 Cependant, l'utilisation actuelle des FPGA est limitée le plus souvent à une seule configuration implantée

alors qu'il est souhaitable d'effectuer des reconfigurations en cours d'exécution. On a donc une répartition spatiale de l'application à réaliser et non temporelle : la capacité du FPGA à se reconfigurer pour
5 réaliser une autre application n'est pas exploitée. Deux raisons principales expliquent cette absence de reconfiguration dynamique : tout d'abord, la majorité des FPGA n'autorise qu'une reconfiguration totale du composant, ce qui entraîne un coût très lourd en terme
10 de temps (typiquement quelques millisecondes à quelques centaines de millisecondes) car il faut re-router l'application pour câbler les opérateurs de façon optimum ; ensuite, du fait que le grain des FPGA est très fin pour pouvoir s'adapter à un grand nombre
15 d'applications, le câblage des opérations a l'inconvénient d'être complexe et de nécessiter une capacité de transmission importante pour un grand nombre de bits (au niveau des bus de contrôle).

Le brevet US6,150,839 divulgue un nouveau
20 type de FPGA qui possède deux niveaux de cache pour pouvoir se reconfigurer partiellement. L'architecture de ce FPGA se présente sous la forme de deux ensembles de cellules qui sont associées respectivement à deux mémoires cache, chacune de ces mémoires pouvant
25 contenir un ou plusieurs contextes pour l'ensemble des cellules. Ce type de FPGA présente plusieurs inconvénients : la reconfiguration des cellules se fait par paquets de cellules et le chargement d'une nouvelle configuration nécessite plusieurs cycles d'horloge ; de
30 plus, la structure de communication inter-cellules est complexe et fait intervenir des bus globaux, les

possibilités très étendues d'interconnexion qui en résultent font que les ressources d'interconnexion prédominent sur celles de traitement.

L'architecture dite DPGA [1,2] (pour
5 'Dynamically Programmable Gate Array'), développée au MIT dès 1994, a pour grain fin une cellule assez simple comportant une LUT ('Look-Up Table' ou table de conversion) à quatre entrées et une bascule programmable. Ces cellules sont regroupées par blocs de
10 16 au sein d'une matrice 4x4. La mise en tableau de ces éléments constitue le DPGA. Deux réseaux d'interconnexions cohabitent dans cette architecture. Le premier réseau est local au bloc de cellules : chaque cellule peut utiliser en entrée les sorties des
15 cellules appartenant à la même ligne ou à la même colonne (soit 6 bits), il est aussi possible que la cellule utilise sa propre sortie comme entrée. Le second réseau permet de fournir à chaque bloc des signaux globaux. En effet, les blocs communiquent entre
20 eux en utilisant des 'crossbar' (réseau d'interconnexion) qui autorisent chaque cellule à recevoir deux signaux globaux par matrice voisine (soit 8 bits, car chaque matrice contient 16 cellules et peut donc fournir 16 bits à ses voisins). Pour les entrées
25 de la LUT de chaque cellule, il y a donc le choix parmi 15 bits : sa sortie, les 6 bits des cellules voisines, les 8 bits des matrices voisines. Le DPGA est capable de mémoriser quatre contextes (ou configurations), ce qui autorise le passage rapide de l'un à l'autre. Ce
30 composant présente toutefois certains inconvénients :

- la structure d'interconnexion est trop complexe pour supporter des applications de type flot de données (par exemple des applications multimédia, interactives ou de cryptographie) ;
- 5 - la simplicité de commande des contextes (deux fils pour le composant) limite les possibilités de réalisation du composant au niveau de la reconfiguration ;
- le maintien des résultats de traitement a lieu à la
- 10 sortie de la cellule, ce qui oblige à maintenir toutes les entrées, du producteur jusqu'au consommateur final ;
- un seul numéro de configuration est distribué pour tout le composant.

15 L'architecture PipeRench [3,4] a été développée pour utiliser la reconfiguration en mode pipeline afin de réaliser des applications de type flot de données. C'est une architecture à gros grain, elle est basée sur des cellules relativement complexes

20 (grain fin) PE (pour 'Processing Element') organisées en 'stripes' ou étages de cellules (gros grain). Ces stripes physiques (câblés sur silicium) sont organisés en lignes et sont reliés entre eux par des interconnexions qui permettent la réalisation d'un

25 anneau ; c'est cet anneau physique qui permet le pipeline. PipeRench découpe une application à réaliser en un certain nombre d'opérations élémentaires qui sont ensuite réparties dans des stripes virtuels (étages virtuels décrits en mémoire). Du fait de la taille

30 importante des cellules PE, il n'est en général pas possible d'avoir autant de stripes physiques que

nécessaire sur le composant : PipeRench virtualise donc les ressources matérielles nécessaires en reconfigurant les stripes physiques de manière à ce qu'ils réalisent successivement toutes les opérations de l'application.

5 Chacune des étapes de l'application correspond à une configuration particulière d'un étage physique de l'architecture. Les données se déplacent d'étage en étage en réalisant les différentes étapes du traitement, ce qui rend difficile la réalisation de

10 traitements non déterministes pour lesquels l'application est dépendante des données. De plus, tout les étages de traitement n'étant pas forcément présents dans l'architecture physique, chaque étage ne doit dépendre que des données de l'étage précédent. Une

15 structure particulière de registre permet cependant de faire transiter des données vers un étage inférieur en appliquant le même pipeline que pour les données classiques ; ceci impose toutefois que des bouclages ne peuvent intervenir qu'au sein d'un même étage (les

20 bouclages entre étages sont prohibés), des registres permettant de réaliser des rétroactions au sein d'un étage. Lorsqu'un étage est déchargé de l'architecture, sa configuration doit être sauvée de façon à maintenir, pour sa réutilisation, l'état de ses variables

25 internes. Le passage des données entre les stripes se fait de manière unidirectionnnelle, c'est pourquoi les configurations sont stockées dans une mémoire unique et sont transférées au coup par coup aux stripes devant être reconfigurés. PipeRench présente encore d'autres

30 inconvénients :

- les stripes sont complexes car ils utilisent des interconnexions locales entre les cellules PE ;
- le nombre très important de bits qu'il faut transmettre pour la remise à jour de la configuration des stripes ;
- les application traitées doivent être du type flot de données et n'avoir que des boucles internes courtes (sur un seul stripe) ;
- les cellules PE (plus élaborées que celles des FPGA) correspondent à un grain encore trop fin pour pouvoir effectuer des traitements complexes ;
- si une application traitée est trop longue par rapport au nombre de stripes , le temps de latence pour un traitement complet peut devenir très important ;
- enfin la consommation du composant est plus élevée d'environ 30% par rapport à celle d'un processeur classique.

Les composants reconfigurables de l'art antérieur présentent donc certains inconvénients ou limitations : les ressources d'interconnexion prédominent sur celles de traitement, leur grain est trop fin, les ressources de routage nécessaires pour le contrôle du composant sont trop importantes, ils ne sont pas adaptables à toutes les applications d'un domaine donné (par exemple : tous les algorithmes symétriques en cryptographie) et ils ne peuvent pas traiter des applications différentes en même temps, ils ne sont pas adaptés pour le traitement de plusieurs types (flot de données ou données dépendantes), leur taux d'utilisation n'est pas élevé notamment lorsque l'on fait des traitements multimodes (pipeline ou

parallèle ou mixte), ils ne sont pas modulables en terme de consommation d'énergie (basse consommation ou haut débit).

5 EXPOSÉ DE L'INVENTION

L'invention vise donc à palier aux inconvénients pré-cités mais aussi à assurer la sûreté structurelle et la robustesse (reconfiguration en cas de dysfonctionnement d'une partie du composant) du
10 composant reconfigurable dynamique.

Afin de pouvoir équilibrer les ressources d'interconnexion et de traitement dans le composant, les opérateurs complexes (au niveau du choix du grain) sont favorisés et les ressources d'interconnexion sont
15 limitées. Les échanges liés au contrôle sont limités en instaurant un nombre de bits de contrôle réduit tout en autorisant des échanges de données à traiter larges. Les ressources de routage global ont été supprimées et ne sont conservées que des ressources locales ou
20 pseudo-locales utiles aux transferts de données. Ce choix est lié à celui du grain de base de l'architecture du composant selon l'invention.

Dans son mode de réalisation de base, le composant à architecture reconfigurable dynamiquement,
25 pour le traitement de données, selon l'invention, est un composant comportant un bloc de traitement de données TD et un contrôleur général CG apte à contrôler le bloc TD caractérisé en ce que :

- le bloc TD comprend une pluralité de blocs
30 élémentaires de traitement de données BE reconfigurables ; chaque bloc élémentaire BE comprenant

deux entrées, E1 et E2, pour la réception de données à traiter, et une sortie S pour la transmission de données traitées ; un bus de données d'entrée commun étant apte à transmettre des données à traiter à
5 l'entrée E1 de chacun des blocs BE ainsi qu'au contrôleur CG ; pour chaque bloc BE, un bus de données de sortie, connecté à sa sortie S, étant apte à transmettre des données traitées vers l'extérieur du composant et, par un bus de données dérivé, à l'entrée
10 E2 d'un seul autre bloc BE ;
- le contrôleur CG est apte à initialiser les configurations des blocs BE et commander leur reconfiguration dynamique, à contrôler les flux de données en sortie de chaque bloc BE de façon à
15 transmettre des données soit vers l'extérieur soit vers l'entrée E2 d'un autre bloc BE, à contrôler les flux de données en entrée de chaque bloc BE.

Ce mode de réalisation de base permet de palier à certains des inconvénients de l'art antérieur, notamment en ce qui concerne la complexité des
20 interconnexions qui est ici fortement réduite. Le grain de la version de base du composant est le bloc élémentaire de traitement BE, il peut être capable de traiter des opérations complexes, et les
25 interconnexions permettent la transmission série ou parallèle, ou la sortie vers l'extérieur, entre chaque paire de blocs BE consécutifs. Le contrôleur CG est bien sûr apte à gérer les problèmes de saturation de chaque bloc BE et il est aussi, classiquement, apte à
30 émettre des requêtes vers l'extérieur pour faire recevoir des données à traiter par le bloc TD (le bloc

TD étant apte à recevoir des données de l'extérieur du composant), à recevoir de l'extérieur des instructions pour le traitement de données par le bloc TD, et les mémoriser et à échanger des signaux de contrôle avec
5 l'extérieur.

Le mode de réalisation de base du composant selon l'invention est illustré, sur un exemple particulier, sur la figure 1 : le composant comporte un bloc de traitement de données TD (1) et un contrôleur
10 général CG (2), le bloc TD peut recevoir des données de l'extérieur (3) et comprend trois blocs élémentaires de traitement de données BE (4,5,6) ; sur le bloc (4) sont indiquées les entrées E1 et E2 ainsi que la sortie S, l'entrée E1 de chaque bloc BE est reliée au bus de
15 données d'entrée commun (7), la sortie S de chaque bloc BE est connectée à un bus de sortie (8,9,10) vers l'extérieur et à l'entrée E2 d'un seul autre bloc par un bus de données dérivé (11,12,13). Les bus de contrôle reliant le contrôleur CG (2) aux blocs BE ou à
20 l'extérieur ne sont pas représentés.

Dans un mode de réalisation à utilisation maximale de bloc, dépendant du mode de base, le contrôleur CG du composant selon l'invention est apte à commander la transmission des données reçues de
25 l'extérieur sur le bus de données d'entrée commun, au fur et à mesure de leur arrivée, successivement à chacun des blocs BE, les données étant transmises à un bloc BE suivant lorsque la capacité maximale de traitement du bloc BE précédent est atteinte.

30 Ce mode de réalisation permet à la fois de gérer la saturation des blocs et d'assurer que chaque

bloc utilisé l'est au maximum de ses capacités de traitement. La capacité maximale de chaque bloc, par exemple, peut être prédéfinie (et donnée au contrôleur CG) ou bien un bloc peut indiquer au contrôleur CG qu'il est saturé en cours de traitement.

Dans un autre mode de réalisation, dit à modes de traitement multiples, du composant selon l'invention, dépendant de l'un ou l'autre des deux modes de réalisation précédents, le contrôleur CG est apte à initialiser les configurations des blocs BE, bloc par bloc, et commander la reconfiguration dynamique des blocs, bloc par bloc, de façon à rendre le bloc TD apte à traiter des données en mode pipeline ou en mode parallèle ou en mode mixte.

Avec cette version du composant il est possible d'effectuer un traitement en mode pipeline, c'est-à-dire en faisant circuler les données de la sortie S d'un bloc à l'entrée E2 du bloc auquel il est connecté par un bus dérivé, ou bien en mode parallèle, c'est-à-dire en utilisant les blocs de manière indépendante (le bloc traite les données transmises par le bus de données d'entrée commun et transmet le résultat en sortie, le bus dérivé n'est alors pas utilisé), ou bien en mode mixte, c'est-à-dire que certains blocs travaillent en parallèle et d'autres en pipeline. Cette version du composant est particulièrement avantageuse pour sa souplesse d'adaptation et aussi parce qu'elle permet de traiter des application différentes, en même temps, sur des blocs BE fonctionnant en mode indépendant, à la

différence de l'art antérieur, grâce à la reconfiguration bloc par bloc.

Un autre mode de réalisation, selon l'invention, d'un composant à architecture reconfigurable dynamiquement, dit à lignes d'unités de traitement, est un composant conforme à l'un quelconque des modes de réalisation précédents et dans lequel :

- chaque bloc BE du bloc TD comprend une pluralité d'unités de traitement de données UT configurables, et correspond à une ligne d'un réseau matriciel d'unités UT dans le bloc TD ; les unités UT de chaque paire d'unités successives dans chaque bloc BE étant aptes à échanger des données via un bloc de communication bidirectionnelle configurable associé BCOM dédié à ces deux unités ;

- l'unité UT d'entrée de chaque bloc BE, qui correspond à la première unité UT sur la ligne correspondant au bloc, est apte à recevoir en entrée des données reçues sur les entrées E1 ou E2 du bloc, et l'unité UT de sortie du bloc, qui correspond à la dernière unité UT sur la ligne correspondant au bloc, est apte à transmettre en sortie des données traitées à la sortie S du bloc ;

- le contrôleur CG est apte, pour chaque bloc BE, à recevoir de l'extérieur des instructions pour le traitement de données par chacune des unités UT du bloc et à les mémoriser, à initialiser les configurations de chaque unité UT et de chaque bloc BCOM du bloc BE et à commander leur reconfiguration dynamique.

Ce mode de réalisation utilise des blocs de communication BCOM, permettant des communications

bidirectionnelles entre unités UT de traitement adjacentes d'un même bloc BE : cette organisation d'un bloc BE autorise une grande souplesse de configuration du bloc (car chaque UT est reconfigurable), notamment

5 pour les traitements multimodes. De plus, il est possible d'utiliser une interconnexion simple, pour les bus de contrôle, entre les unités UT, le blocs BCOM et le contrôleur CG : par exemple, chaque unité UT de chaque bloc BE peut être connectée au contrôleur CG par

10 un bus de contrôle commun pour le bloc BE ; de même, chaque bloc BCOM de chaque bloc BE peut être connecté au contrôleur CG par un bus de contrôle commun pour le bloc BE. L'homme du métier notera que, lors d'un fonctionnement en parallèle, la structure avec un

15 nombre pair d'unités UT sur une ligne est plus efficace : en effet, après le remplissage d'un bloc BE avec des données (du début vers la fin du bloc) et quand le bloc BE est saturé de données, si le nombre d'unités UT est pair et du fait du caractère

20 bidirectionnel des blocs BCOM, les unités peuvent travailler par paires alors que si le nombre d'unités est impair il restera une unité inutilisée. La structure du bloc TD est équivalente à un réseau matriciel : les blocs BE, et leurs unités UT, en

25 forment les lignes ; d'une ligne à l'autre, les unités UT (de rang correspondant sur les lignes) en forment les colonnes.

La figure 2 illustre un exemple de structure d'un bloc BE du composant à lignes d'unités

30 de traitement : le bloc BE (14) comprend 4 unités de traitement UT (15) alternant avec des blocs de

communication BCOM (16) ; l'unité UT d'entrée (15) du bloc est apte à recevoir des données en entrée via un bus de données (18) et l'unité UT de sortie (17) du bloc est connectée en sortie (19) à la sortie S du bloc. Les
5 unités UT sont connectées par un bus de contrôle commun (20) au contrôleur CG (non représenté), et les blocs de communication BCOM sont connectés par un bus de contrôle commun (21) au contrôleur CG.

Un autre mode de réalisation du composant
10 selon l'invention, dépendant du mode de réalisation à lignes d'unités de traitement, est un composant, dit à registres inter-lignes, dans lequel :

- les blocs BCOM de chaque paire de blocs BCOM, dont les unités UT associées sont, selon le réseau matriciel
15 d'unités du bloc TD, sur des colonnes adjacentes et sur des lignes adjacentes, sont aptes à échanger des données par un registre inter-lignes associé REG, de façon à permettre l'échange de données entre unités UT d'une ligne à l'autre;
- 20 - le contrôleur CG est apte à commander la reconfiguration dynamique des unités UT et des blocs BCOM du bloc TD pour rendre le bloc TD apte à traiter des données en mode à lignes dépendantes, pour au moins deux lignes adjacentes du réseau matriciel, en
25 commandant la bifurcation des données entre deux blocs BE correspondant aux deux lignes adjacentes, via un registre inter-lignes REG entre ces deux lignes.

Le composant à registres inter-lignes est particulièrement avantageux en ce qu'il permet de
30 disposer d'un mode de traitement supplémentaire par rapport à l'art antérieur : le mode à lignes

dépendantes (ou, de manière équivalente, à blocs BE dépendants). Ceci accroît encore les possibilités de traitement et permet d'améliorer le taux d'utilisation des ressources du composant ainsi que le débit de données. Par exemple si l'on dispose de deux lignes capables de traiter chacune 256 bits de données, le passage en mode dépendant entre ces lignes permet de traiter 512 bits de données. L'homme du métier notera qu'avec une architecture PipeRench, à chaque coup d'horloge, on peut au mieux avec deux stripes réaliser une reconfiguration d'un stripe et une exécution sur l'autre ; ceci implique qu'une seule donnée peut être traitée à la fois, alors qu'avec l'architecture selon l'invention on peut traiter deux données avec deux unités adjacentes car durant les transferts d'unité à unité, les unités sont reconfigurées.

L'invention concerne aussi un composant avec contrôle de registre, qui est un mode de réalisation dépendant du mode précédent à registres interlignes, dans lequel le contrôleur CG est apte à transmettre, via un bus de contrôle, des signaux de type RESET ou ENABLE ou d'horloge à chaque registre REG du bloc TD, et dans lequel chaque registre inter-lignes REG entre deux blocs BCOM est apte à :

- stocker des données reçues d'un bloc BCOM dans une mémoire et les rendre disponibles pour l'autre bloc BCOM ;
- recevoir des signaux de type RESET ou ENABLE ou d'horloge émis par le contrôleur CG et exécuter les commandes correspondantes à ces signaux.

Ce contrôle de registre inter-lignes, pour le stockage en vue de la transmission de données entre deux blocs de communication, activé par la commande ENABLE, ou pour la remise à zéro de la mémoire de
5 stockage, activée par la commande RESET, permet de faciliter la gestion des transferts par le contrôleur CG.

Un autre mode de réalisation de l'invention, dépendant de l'un quelconque des modes
10 précédents, concerne un composant avec multiplexage dans lequel : les entrées E1 et E2 d'un bloc BE sont celles d'un multiplexeur associé au bloc, la sortie du multiplexeur étant connectée à l'entrée de l'unité UT d'entrée du bloc BE pour la transmission de données à
15 cette unité.

Cette version du composant permet donc de simplifier les connexions en n'ayant qu'une seule entrée pour l'unité UT d'entrée de bloc BE, pour recevoir des données des entrées E1 ou E2 du bloc.

20 Le mode de réalisation de l'invention, dépendant de l'un quelconque des modes de réalisation du composant à lignes d'unités de traitement ou à registres inter-lignes ou avec contrôle de registre ou avec multiplexage, concerne un composant avec banc de
25 cellules dans lequel : les unités de traitement de données UT du bloc TD sont des bancs de cellules élémentaires synchrones de traitement de données CE configurables, chaque banc comportant au moins une cellule élémentaire CE ; les cellules CE de toute unité
30 UT d'un bloc BE quelconque sont connectées à un bus de contrôle commun qui les relie au contrôleur CG, le

contrôleur CG étant apte à contrôler les cellules CE et à commander leur reconfiguration; pour chaque paire d'unités UT adjacentes dans un bloc BE, les cellules élémentaires d'une unité UT sont aptes à échanger des données avec les cellules élémentaires de l'autre unité UT via le bloc BCOM associé à ces unités, le bloc BCOM étant apte à réaliser ces échanges de données ; chaque cellule CE de l'unité UT d'entrée de chaque bloc BE, qui correspond à la première unité UT sur la ligne correspondant au bloc, est apte à recevoir en entrée des données reçues en entrée de l'unité UT par un bus de données commun à toute cellule CE de l'unité UT; chaque cellule CE de l'unité UT de sortie de chaque bloc BE, qui correspond à la dernière unité UT sur la ligne correspondant au bloc, est apte à transmettre en sortie des données à la sortie S du bloc par un bus de données commun à toute cellule CE de l'unité UT.

Cette version du composant, dans laquelle la cellule synchrone configurable CE est le grain fin de l'architecture (bien qu'il s'agisse d'un gros grain par rapport aux architectures de l'art antérieur), permet de paralléliser localement le travail des cellules élémentaires CE d'une unité UT, tout en ayant des interconnexions réduites puisque les cellules CE d'une même unité de traitement n'échangent pas de données entre elles. Une unité UT (ou banc) comporte en général n cellules CE et, si k désigne la taille des opérandes manipulés par les cellules élémentaires, l'unité UT est apte à traiter des blocs de p bits, avec $p = n * k$. La figure 3 illustre un banc UT de cellules CE (22), comportant ici 8 cellules, et les connexions

de chacune de ces cellules avec un bus de données d'entrée (23) pour la transmission de données à traiter, et avec un bus de données de sortie (24) pour la transmission de données traitées ; les cellules CE
5 sont connectées au contrôleur CG (non représenté) par un bus de contrôle commun (25).

La figure 4 illustre, pour un bloc BE, un exemple de connexions entre deux bancs de cellules CE adjacents au sein du bloc via un bloc BCOM : le bloc
10 BCOM (26) permet la transmission de données entre une cellule CE (28) d'une première unité UT et les cellules (29), (30) et (31) d'une deuxième unité UT selon des signaux de commande reçus du contrôleur CG par ces cellules, via le bus de contrôle des cellules (32), et
15 par le bloc BCOM, via le bus de contrôle (33) ; la cellule (35) recevant des données de la cellule (34). Ceci illustre les transferts bidirectionnels (27) permis par le bloc BCOM entre cellules de bancs (ou unités) différents, alors que dans une même unité UT
20 les cellules CE ne communiquent pas (par exemple les cellules (28) et (35)).

L'invention concerne aussi un mode de réalisation du composant, dit avec contrôle local de communication, dépendant du mode précédent avec banc de
25 cellules, dans lequel chaque bloc BCOM, associé à des unités UT, comporte :

- un bloc de transmission BT apte à réaliser des échanges de données entre les cellules élémentaires CE d'une unité UT et les cellules CE de l'autre unité UT
30 auxquelles BCOM est associé ; le bloc BT étant apte à être configuré par un contrôleur local de bloc CLB;

- un contrôleur local de bloc CLB apte à recevoir et interpréter des signaux de contrôle provenant du contrôleur CG et à configurer le bloc BT, selon des données de configurations reçues et stockées dans une
5 mémoire de configuration locale MB, pour la transmission de données ; le contrôleur CLB étant apte à gérer le chargement de la mémoire MB en fonction de signaux de contrôle reçus du contrôleur CG ;
- un contrôleur général CG apte à contrôler les
10 contrôleurs locaux CLB du bloc TD.

Ce composant avec contrôle local de communication présente de nouveaux avantages : les blocs de communication BCOM sont en effet reconfigurables et il est possible de choisir les
15 cellules élémentaires CE à interconnecter (entre unités UT adjacentes d'un bloc BE), ce degré de liberté supplémentaire permettant encore plus de souplesse de traitement et d'efficacité d'utilisation des cellules (notamment pour le mappage des opérateurs élémentaires
20 dans les cellules). Les communications entre bancs, via les blocs BCOM, sont bidirectionnelles, ce qui est un avantage sur les composants de type PipeRench où le passage des données entre stripes est unidirectionnel. A titre d'exemple, le contrôleur CLB comprendra
25 typiquement, outre la mémoire vive MB (ou 'RAM'), une machine à états finis (ou FSM, 'Finite State Machine') de contrôle.

Dans un autre mode de réalisation du composant selon l'invention, avec contrôle local de
30 cellule, dépendant du mode de réalisation avec banc de cellules ou du mode avec contrôleur local de

communication, chaque cellule élémentaire synchrone CE d'un banc comporte :

- un bloc opérateur BO apte à recevoir des données en entrée par un bus de données d'entrée, à les stocker et
5 à effectuer un traitement sur ces données selon des signaux de configuration provenant d'un contrôleur local de cellule CLC ; à transmettre le résultat du traitement en sortie par un bus de données de sortie;
- un contrôleur local de cellule CLC apte à :
10
 - recevoir des données par le bus de données d'entrée ;
 - gérer les différents modes de fonctionnement de la cellule CE, c'est-à-dire un mode d'initialisation, pour le chargement des mots de
15 configuration de la cellule dans une mémoire de configuration locale MC, un mode de test, pour effectuer des tests structurels de la cellule, et un mode normal pour l'exécution des opération de traitement de données, en fonction de signaux de
20 configuration émis par le contrôleur CG et reçus par le contrôleur local via le bus de contrôle commun ;
 - gérer le chargement de la mémoire MC en fonction d'un mode de fonctionnement de la cellule indiqué
25 par le contrôleur CG ;
 - interpréter des signaux de configuration reçus du contrôleur CG pour soit charger un mot de configuration correspondant dans un registre de configuration local RCL pour le maintenir pendant
30 le cycle de traitement de données par la cellule, soit configurer le bloc BO en fonction d'un mot de

configuration figurant dans le registre RCL ;
- effectuer des tests structurels de la cellule CE
sur réception d'un signal de mode test émis par le
contrôleur CG ;

- 5 - un contrôleur général CG apte à contrôler les
contrôleurs locaux CLC du bloc TD.

Dans ce composant avec contrôle local de
cellule, les cellules CE d'un banc sont
reconfigurables. L'efficacité de traitement est donc
10 accrue et notamment il est possible de traiter
plusieurs applications. Le bloc TD est alors apte à
traiter ces applications différentes, y compris de
manière simultanée. De plus, il est possible
d'effectuer des tests sur les cellules et ainsi
15 détecter des défaillances. L'homme du métier notera que
la stratégie de fonctionnement du composant diffère de
celle de l'art antérieur dans lequel les 'cellules'
voisines peuvent communiquer entre elles, ici on
profite d'une reconfiguration de cellules pour réaliser
20 des échanges de données entre les cellules (de bancs
différents). La stratégie de contrôle, quant à elle,
repose sur un contrôle à deux niveaux : celui du
contrôleur général CG au niveau haut de l'architecture
et celui du contrôle local tant au niveau du grain fin
25 (contrôleurs CLC des cellules CE) qu'au niveau des
blocs de communication (contrôleurs CLB). Par rapport
au FPGA du brevet US6,150,839, dans lequel la
reconfiguration se fait par blocs de cellules, il est
ici possible de reconfigurer cellule par cellule ; de
30 plus l'architecture selon l'invention peut se
reconfigurer en un cycle d'horloge au lieu plusieurs

cycles nécessaires (du fait du temps nécessaire au chargement de la nouvelle configuration dans un bloc) pour ce FPGA. A titre d'exemple de réalisation, le bloc BO comprendra typiquement une LUT, une mémoire vive et
5 une unité arithmétique logique (ou ALU, 'Arithmetic Logic Unit').

Dans un autre mode de réalisation d'un composant selon l'invention, ou composant à configuration par numéro, qui dépend du mode de
10 réalisation précédent avec contrôle local de cellule, le contrôleur local de cellule CLC est apte à interpréter des signaux de configuration reçus du contrôleur CG pour d'abord lire un numéro de configuration de cellule sur le bus de contrôle commun
15 puis charger un mot de configuration correspondant localement à ce numéro, selon une table de correspondance locale, dans le registre RCL ; les numéros de configuration de cellule étant stockés dans une mémoire du contrôleur CG et le contrôleur CLC étant
20 apte à mémoriser la table de correspondance locale.

Ce mode de réalisation a l'avantage de requérir beaucoup moins de ressources de transmission, et donc de simplifier la connectique du composant. En effet, la transmission d'un simple numéro de
25 configuration nécessite un bus de contrôle beaucoup moins large que celui nécessaire pour la transmission d'un mot de configuration ; ceci étant possible grâce à la table de correspondance locale. De plus, la correspondance locale entre un numéro et un mot de
30 configuration introduit un degré de liberté intéressant : un même numéro peut en effet correspondre

à des mots de configuration différents dans des cellules différentes, puisque les tables de correspondance sont locales.

L'invention concerne aussi un mode de
5 réalisation du composant, dit avec BYPASS, dépendant du mode de réalisation précédent à configuration par numéro ou du mode avec contrôle local de cellule, dans lequel le contrôleur CG est apte à recevoir d'une cellule CE une indication de dysfonctionnement
10 résultant d'un test structurel mis en œuvre sur cette cellule, et à commander au contrôleur local CLC de la cellule CE de configurer la cellule en mode BYPASS.

Ce mode de réalisation avec BYPASS (ou dérivation) permet d'obtenir un composant robuste au
15 niveau des cellules CE.

De même, il est possible d'obtenir la robustesse au niveau des blocs BCOM grâce à un mode de réalisation d'un composant selon l'invention, dit avec BYPASS de communication, qui dépend du mode de
20 réalisation précédent avec BYPASS et du mode avec contrôle local de communication, dans lequel le contrôleur local de bloc CLB d'un bloc de communication BCOM, connecté à une unité UT, est apte à configurer le bloc BCOM en mode BYPASS lorsque le contrôleur CG,
25 ayant reçu d'une ou plusieurs cellules CE de l'unité UT une indication de dysfonctionnement, transmet au contrôleur CLB un signal de passage en mode BYPASS ; le contrôleur CG étant apte à commander le passage en mode BYPASS d'un bloc BCOM d'une unité UT lorsqu'une
30 commande de configuration en mode BYPASS d'une cellule

CE de l'unité UT a été transmise au contrôleur local CLC de la cellule CE.

Avec le mode de réalisation avec by-pass de communication, un fonctionnement dégradé du composant
5 (c'est-à-dire avec perte d'unités UT ou de blocs BCOM) est donc permis. L'homme du métier notera que la possibilité de commander des cellules ou des blocs BCOM en mode BYPASS offre notamment celle de disposer, dans un bloc BE, d'unités UT ou de blocs BCOM
10 'supplémentaires' qui ne sont pas utilisés en l'absence de dysfonctionnement mais qui peuvent être activés (via le contrôleur CG) en cas de dysfonctionnement dans le bloc ; ceci permettant de renforcer encore la robustesse du composant.

15 Un mode de réalisation d'un composant selon l'invention, dit avec registre local, permet d'augmenter la capacité de traitement du composant. Dans ce mode de réalisation, dépendant de l'un quelconque des modes avec banc de cellules ou à
20 contrôle local de communication ou à contrôle local de cellule ou à configuration par numéro ou avec by-pass ou avec by-pass de communication, le résultat du traitement de données effectué par une cellule CE est apte à être stocké dans un registre local ; un résultat
25 stocké dans ce registre local étant maintenu lors d'une reconfiguration de la cellule.

Le composant avec registre local permet donc de sauvegarder les données calculées pendant une reconfiguration et un autre avantage lié à cette
30 sauvegarde, et cela même si le bloc BE du composant est 'minimal' c'est-à-dire réduit à deux unités UT et un

bloc BCOM, est que grâce au bloc BCOM à communication bidirectionnelle, il est possible de traiter deux données en même temps au niveau du bloc BE et transmettre les résultats, alors que dans une
5 architecture PipeRench on ne peut traiter qu'une donnée avec deux stripes (pendant qu'un stripe effectue le traitement, l'autre se reconfigure), et ceci suppose bien sûr que les données soient sauvegardées pendant la reconfiguration. Un autre avantage important dû à ce
10 registre local est de permettre l'échange de données stockées entre deux unités UT pendant la phase de reconfiguration, ce qui contribue à masquer le cycle de reconfiguration.

Un autre mode de réalisation d'un composant
15 selon l'invention, dépendant de l'un quelconque des modes de réalisation précédents de l'invention, concerne un composant avec gestion d'entrée-sortie comportant :

- un bloc de gestion d'entrée GE apte à recevoir des
20 données de l'extérieur et à stocker temporairement ces données reçues, à formater des données stockées selon le mode d'initialisation ou le mode normal indiqué par le contrôleur CG via un bus de contrôle propre, à transmettre des données formatées à l'entrée E1 des
25 blocs BE du bloc TD via le bus de données d'entrée commun ;
- un bloc de gestion de sortie GS, connecté au contrôleur CG par un bus de contrôle propre pour la commande de réception de données traitées, connecté au
30 bus de données de sortie de chacun des blocs BE du bloc TD pour recevoir des données traitées, et apte à

reformater des données traitées reçues, à stoker dans une mémoire tampon des données reformatées, à recevoir une requête de transmission depuis l'extérieur, à transmettre à l'extérieur des données stockées dans la
5 mémoire tampon sur requête reçue de l'extérieur.

Cette version du composant, avec gestion d'entrée-sortie, facilite les échanges des données avec l'extérieur (en particulier avec le stockage temporaire de données) et donc l'utilisation du composant par
10 divers dispositifs. L'homme du métier peut remarquer que le gestionnaire d'entrée GE permet une simplification de l'implantation d'une application à traiter (ou mappage de l'application) sur le composant.

La figure 5 illustre un exemple d'un
15 composant avec gestion d'entrée-sortie : un gestionnaire d'entrée GE (36) qui reçoit des données de l'extérieur par un bus de données (43) et transmet des données formatées à trois blocs BE (40,41,42) par un bus de données commun (44) ; les bus de données de
20 sortie des blocs BE (45,46,47) sont connectés au gestionnaire de sortie GS (37) qui transmet à l'extérieur des données reformatées par un bus de données (48) ; le contrôleur général CG est connecté à GE par un bus de contrôle (38), à GS par un bus de
25 contrôle (39), aux blocs BE1, BE2 et BE3, respectivement, par les bus de contrôle (49,50,51).

L'homme du métier sait qu'un composant reconfigurable doit pouvoir être testé régulièrement. Une méthode bien connue pour cela est la méthode JTAG
30 [5], normalisée par IEEE, qui est aussi utilisée pour le chargement (ou le rechargement) de mots de

configuration grâce à la possibilité offerte par la norme IEEE de disposer d'instructions spécifiques à un utilisateur; le composant doit alors intégrer un contrôleur TAP et des bus de contrôle spécifiques.

5 C'est pourquoi un autre mode de réalisation d'un composant selon l'invention, dépendant de l'un quelconque des modes de réalisation du composant avec contrôle local de cellule ou à configuration par numéro ou avec BYPASS ou avec BYPASS de communication ou avec
10 registre local ou avec gestion d'entrée-sortie, est un composant avec JTAG dans lequel le contrôleur local CLC de cellule CE, le contrôleur local CLB de bloc BCOM et le contrôleur général CG sont aptes à mettre en œuvre une méthode JTAG pour le chargement des configurations
15 en phase de reconfiguration dynamique du composant, et pour la réalisation de tests ; le contrôleur CG comprenant un contrôleur TAP connecté par des bus série JTAG, bloc BE par bloc BE, aux cellules CE et aux blocs BCOM du bloc TD.

20 Le choix du protocole JTAG, appliqué au niveau local, est tout à fait compatible avec le choix du contrôle à deux niveaux déjà évoqué. Ce dernier s'explique par le fait que l'on veut pouvoir reconfigurer les cellules en un temps très court (une
25 période d'horloge au maximum) : une reconfiguration au plus proche de l'opérateur de base est le plus efficace car on minimise l'échange d'information et donc son interprétation. Un des avantages de ce choix étant qu'il permet de réduire de façon considérable les
30 ressources de routage pour le contrôle (un bus pour quelques bits de contrôle au lieu d'un bus de contrôle

de la largeur d'un mot de configuration), et cette économie importante permet de diminuer la contrainte pour les bus de données.

La figure 6 est une illustration d'un
5 exemple d'insertion d'une chaîne JTAG dans un ensemble,
au sein d'un bloc BE, comprenant deux bancs de cellules
CE et le bloc BCOM de communication associé : la chaîne
JTAG (52) est connectée successivement aux cellules CE
du premier banc (53), au bloc BCOM (54) et aux cellules
10 CE du deuxième banc (55); cette connexion de contrôle
vient en plus des bus de contrôle (sans JTAG) des
cellules (56) et du bus de contrôle du bloc BCOM (57)
qui sont connectés au contrôleur CG (non représenté).

Un autre composant selon l'invention, est
15 un composant à gros grain, conforme à l'un quelconque
des modes de réalisation de l'invention à lignes
d'unités de traitement ou à registres inter-lignes ou
avec contrôle de registre ou avec multiplexage ou avec
banc de cellules ou avec contrôle local de
20 communication ou avec contrôle local de cellule ou à
configuration par numéro ou avec BYPASS ou avec BYPASS
de communication ou avec registre local ou avec gestion
d'entrée-sortie ou avec JTAG, dans lequel, sur les
lignes du réseau, les unités UT sont en nombre pair N
25 et les blocs BCOM sont répartis dans N/2 cellules de
base CB, chaque cellule CB comportant deux unités UT
adjacentes et le bloc BCOM associé à ces unités ; le
contrôleur CG étant apte à commander la reconfiguration
dynamique des cellules CB du réseau et des blocs BCOM
30 de communication entre cellules CB, ligne par ligne,

pour rendre le bloc TD apte à traiter des données au moyen des cellules CB.

Cette version du composant selon l'invention fait apparaître un organe très important en matière de traitement de données qui est la cellule de base CB. Ainsi on dispose d'une architecture reconfigurable à trois niveaux hiérarchiques : celui de la cellule élémentaire CE (grain fin) pour l'exécution synchrone d'opérations élémentaires, celui de la cellule de base (gros grain) permettant de réaliser des traitements complexes, et enfin celui du composant lui-même. Le nombre plus ou moins grand de cellules de base configurées pour exécuter un traitement de données détermine alors le caractère, respectivement, à haut débit de données ou à basse consommation d'énergie du composant. Ce caractère, adaptable suivant des signaux de commande (au niveau du contrôleur CG), combiné aux grandes possibilités de configuration (ou de reconfiguration) de chaque cellule CB (via celles des cellules CE et du bloc BCOM) permet d'atteindre les buts visés par l'invention.

Enfin, l'invention concerne un dernier mode de réalisation du composant, dépendant du mode de réalisation à gros grain, dit à gros grain autonome, pour le traitement d'un ensemble d'applications comportant au moins une application, dans lequel chaque cellule de base CB du bloc TD configurée pour effectuer un traitement est apte à réaliser les opérations nécessaires au traitement complet d'au moins une application de l'ensemble, chaque application de

l'ensemble étant apte à être réalisée complètement par .
au moins une cellule de base CB.

Ce composant à gros grain autonome est la
version optimale du composant à architecture
5 reconfigurable de manière dynamique selon l'invention.
Il est capable de traiter un ensemble d'applications
(qui peut se réduire à une seule application) grâce à
des cellules de base autonomes quant au traitement de
ces applications. Cette autonomie des cellules CB
10 permet de bénéficier de manière optimale des
possibilités très souples de reconfiguration du
composant et d'utilisation des divers modes de
traitement en pipeline, en parallèle, mixte ou en
lignes dépendantes ou de leurs diverses combinaisons.

15

BRÈVE DESCRIPTION DES DESSINS

Les caractéristiques et avantages de
l'invention exposée plus haut, ainsi que d'autres qui
ressortiront de la description suivante de modes
20 particuliers de réalisation, donnés à titre d'exemples,
apparaîtront davantage en se référant aux dessins
annexés, sur lesquels :

- la figure 1, décrite plus haut, illustre le mode
de réalisation de base du composant selon
25 l'invention, dans un cas particulier avec trois
blocs élémentaires BE ;
- la figure 2, décrite plus haut, illustre un
exemple de structure d'un bloc BE du composant à
lignes d'unités de traitement ;

- la figure 3, décrite plus haut, illustre une unité de traitement UT avec cellules CE du composant avec banc de cellules ;
- 5 - la figure 4, décrite plus haut, illustre, pour un bloc BE, un exemple de connexions entre deux bancs de cellules CE adjacents au sein du bloc BE via un bloc BCOM d'un composant avec banc de cellules ;
- la figure 5, décrite plus haut, illustre un exemple d'un composant avec gestion d'entrée-sortie ;
- 10 - la figure 6, décrite plus haut, est une illustration d'un exemple d'insertion d'une chaîne JTAG dans une partie d'un composant avec JTAG ;
- la figure 7 est un schéma d'une vue générale d'un composant à gros grain autonome ;
- 15 - la figure 8 est un schéma illustrant la phase de remplissage d'une ligne de cellules de base en mode parallèle ;
- la figure 9 est un schéma illustrant la phase de bouclage d'une ligne de cellules de base en mode parallèle ;
- 20 - la figure 10 est un schéma illustrant la phase de vidage d'une ligne de cellules de base en mode parallèle ;
- 25 - la figure 11 est un schéma résumant les étapes de fonctionnement en mode parallèle d'un composant à gros grain autonome.

EXPOSÉ DÉTAILLÉ D'UN MODE DE RÉALISATION PARTICULIER

30 Le fonctionnement d'un exemple de réalisation d'un composant à gros grain autonome selon

l'invention comporte plusieurs phases qui peuvent correspondre à des stratégies d'utilisation différentes. Parmi toutes ces phases, la mise en œuvre des phases suivantes (à titre d'exemple) sera
5 détaillée : celles d'initialisation, de fonctionnement normal d'un fonctionnement en parallèle.

La figure 7 illustre l'exemple de composant retenu : un composant à deux lignes (ou deux blocs BE), la ligne d'indice 0 et la ligne d'indice 1,
10 avec sur chaque ligne deux cellules de base CB et leurs unités de traitement UT indicées par un couple (i,j) ou i est l'indice de ligne et j celui de colonne. Ainsi, sur la ligne 0, la cellule CB(0,1) (60) comprend les unité UT(0,1) et UT(0,2) et leur bloc de communication
15 associé BCOM(0,1) (63) pour lequel les indices sont, par convention, ceux de l'unité UT précédant le bloc sur la ligne ; la seconde cellule CB(0,2) (62) sur la ligne communique avec la première via le bloc BCOM(0,2) (64) et comprend les unités UT(0,3) et UT(0,4) et le
20 bloc BCOM(0,3) (65). De même, la seconde ligne comprend les cellules de base CB(1,1) (66) et CB(1,2) (67), respectivement formées de UT(1,1), BCOM(1,1) (68), UT(1,3) et UT(1,3), BCOM(1,3) (69), UT(1,4), qui communiquent via le bloc BCOM(1,2) (70). Des registres
25 interlignes (71,72,73) font respectivement communiquer les blocs BCOM(0,1) (63) et BCOM(1,1) (68), BCOM(0,2) (64) et BCOM(1,2) (70), BCOM(0,3) (65) et BCOM(1,3) (67). Le composant comporte un bus de données d'entrée du bloc TD (74), un gestionnaire d'entrées GE (75)
30 connecté au bus de données d'entrée commun (76), un bus de données de sortie du bloc TD (77) connecté au

gestionnaire de sortie GS (78) qui reçoit les bus de données de sortie respectivement des lignes 0 (79) et 1 (80). Le contrôleur général CG (81) relié : au bus de contrôle commun des unités UT de la ligne 0 (82) et au
5 bus de contrôle commun des blocs BCOM de la ligne 0 (83), au bus de contrôle commun des unités UT de la ligne 1 (84) et au bus de contrôle commun des blocs BCOM de la ligne 1 (85) ; le contrôleur CG étant apte à échanger des signaux de contrôle avec l'extérieur (86)
10 ou avec un registre à décalage périphérique (87) ('Boundary Scan') pour le contrôle JTAG.

a) la phase d'initialisation

Lors de cette phase, on charge les
15 différentes données de configuration utiles dans les différentes mémoires. Cette phase existe après un reset de mise sous tension ou après un reset logiciel provoqué. Le chargement se fait par le bus de données (76) (chargement en parallèle). On charge en premier
20 lieu les configurations des cellules ligne par ligne successivement, puis on procède de même pour les blocs de communication. Ensuite des tests sont effectués sur le composant pour vérifier la bonne configuration initiale ainsi que la sûreté structurelle (par exemple
25 avec un test BIST) : à l'issue de cette étape le contrôleur CG signale à l'extérieur que le composant est prêt à recevoir des données.

b) phase de fonctionnement en parallèle

30 Dans cet exemple, les lignes fonctionnent de façon autonome l'une par rapport à l'autre, et

toutes les étapes décrites ci-après sont synchronisées à l'horloge (indiquée par Clock sur les dessins).

Au niveau de chaque ligne, il y a trois étapes de fonctionnement :

- 5 - une étape 1 de chargement de ligne, par exemple la ligne 0, la ligne étant considérée comme un pipeline que l'on remplit.

L'étape 1 est illustrée, pour le remplissage de la ligne 0 (en référence à la figure 7), sur la figure 8 :
10 en 8a les opérations exécutées par les unités sont indiquées par op1, op2 etc.. ; en 8b est représenté un chronogramme correspondant aux opérations effectuées sur la ligne 0, les impulsions de l'horloge sont représentées en haut du chronogramme (Clock).

15 Le gestionnaire GE récupère des blocs de données, par exemple de 64 bits, et reconstitue le mot de p bits qui sera envoyé sur la première unité UT de la ligne UT(0,1) de la première cellule CB(0,1). On charge UT(0,1) avec la première donnée 1 qui exécute
20 l'opération 1, puis on transfère le résultat vers UT(0,2) ; ensuite UT(0,1) exécute l'opération 1 sur un bloc de données 2 tandis que UT(0,2) exécute l'opération 2 sur le bloc de données 1. On réalise de nouveau le transfert jusqu'à atteindre le bout de la
25 ligne (toutes les unités de traitement sont occupées par une donnée).

- une étape 2 de bouclage (exemple de la ligne 0) : pour réaliser le reste du traitement à effectuer (de l'application) en n'employant que les ressources de la
30 ligne. On travaille alors par cellules de base (autonomes) pour réaliser le traitement en parallèle.

L'étape 2 est illustrée, pour le bouclage de la ligne 0, sur la figure 9 : en 9a les opérations effectuées par les unités de la ligne sont indiquées ; en 9b est représenté un chronogramme de cette étape de bouclage ;
5 la notation 'donnée 4/3' indiquant un échange des données traitées 4 et 3 entre les unités UT(0,1) et UT(0,2).

- une étape 3 de vidage de ligne, qui est illustrée, pour le vidage de la ligne 0, sur la figure 10 : en 10a
10 les opérations effectuées par les unités de la ligne sont indiquées ; en 10b est représenté un chronogramme de cette étape de vidage :

lorsqu'on arrive à la fin de l'algorithme, la cellule de base connectée au bus de sortie CB(0,2) va pouvoir
15 décharger les données traitées, on passe à la phase de vidage du pipeline. En effet, l'unité UT(0,4) de la deuxième cellule décharge le bloc de données 1 qui vient d'être traité dans le gestionnaire GS. Cependant, le bloc de données 2, dans UT(0,3), n'a pas encore subi
20 la dernière opération. Il passe donc dans UT(0,4) pour subir cette opération, l'unité UT(0,2) transférant son bloc de données 3 à UT(0,3) etc., jusqu'au vidage complet de la ligne 0.

Dès que le gestionnaire de sortie a une
25 nouvelle donnée traitée, il le signale à l'extérieur pour sa récupération. Ce schéma se répète, en parallèle, pour la ligne 1.

La figure 11 résume les étapes précédentes et indique les traitements effectués en parallèle sur
30 les lignes 0 et 1 du composant représenté en 11a. Les

étapes effectués sur chacune de ces lignes, au cours du temps, sont représentées sur un chronogramme en 11b.

RÉFÉRENCES

- 5 [1] : E.Tau, I.Eslick, D.Chen, J.Brown, A.DeHon : 'A First Generation DPGA Implementation', in Proceedings of the Third Canadian Workshop on Field-Programmable Devices, pages 138-143, May 1995.
- 10 [2] : André DeHon : 'Reconfigurable Architectures for General-Purpose Computing', Ph.D. Thesis, MIT, August 1996.
- 15 [3] : S.Copen Goldstein, H.Schmit, M.Budiu, S.Cadambi, M.Moe and R.Taylor : 'PipeRench : A Reconfigurable Architecture and Compiler', in IEEE Computer, vol.33, N°4, April 2000.
- 20 [4] : Benjamin A.Levine and Herman H.Schmit : 'PipeRench : Power and Performance Evaluation of a Programmable Pipelined Datapath', Hot Chips 14, Palo Alto, CA, August 2002.
- 25 [5] : JTAG; Test Technology Standards Committee "IEEE Std. 1149.1 Standard Test Access Port and Boundary-Scan Architecture", Institute of Electrical and Electronics Engineers, October 21, 1993.

REVENDICATIONS

1. Composant à architecture reconfigurable dynamiquement pour le traitement de données comportant
- 5 un bloc de traitement de données TD et un contrôleur général CG apte à contrôler le bloc TD caractérisé en ce que :
- le bloc TD comprend une pluralité de blocs élémentaires de traitement de données BE
 - 10 reconfigurables ; chaque bloc élémentaire BE comprenant deux entrées, E1 et E2, pour la réception de données à traiter, et une sortie S pour la transmission de données traitées ; un bus de données d'entrée commun étant apte à transmettre des données à traiter à
 - 15 l'entrée E1 de chacun des blocs BE ainsi qu'au contrôleur CG ; pour chaque bloc BE, un bus de données de sortie, connecté à sa sortie S, étant apte à transmettre des données traitées vers l'extérieur du composant et, par un bus de données dérivé, à l'entrée
 - 20 E2 d'un seul autre bloc BE ;
 - le contrôleur CG est apte à initialiser les configurations des blocs BE et commander leur reconfiguration dynamique, à contrôler les flux de données en sortie de chaque bloc BE de façon à
 - 25 transmettre des données soit vers l'extérieur soit vers l'entrée E2 d'un autre bloc BE, à contrôler les flux de données en entrée de chaque bloc BE.

2. Composant selon la revendication 1 dans
- 30 lequel le contrôleur CG est apte à commander la transmission des données reçues de l'extérieur sur le

bus de données d'entrée commun, au fur et à mesure de leur arrivée, successivement à chacun des blocs BE, les données étant transmises à un bloc BE suivant lorsque la capacité maximale de traitement du bloc BE précédent est atteinte.

3. Composant selon la revendication 1 ou 2 dans lequel le contrôleur CG est apte à initialiser les configurations des blocs BE, bloc par bloc, et commander la reconfiguration dynamique des blocs, bloc par bloc, de façon à rendre le bloc TD apte à traiter des données en mode pipeline ou en mode parallèle ou en mode mixte.

4. Composant selon l'une quelconque des revendications 1 à 3 dans lequel :

- chaque bloc BE du bloc TD comprend une pluralité d'unités de traitement de données UT configurables, et correspond à une ligne d'un réseau matriciel d'unités UT dans le bloc TD ; les unités UT de chaque paire d'unités successives dans chaque bloc BE étant aptes à échanger des données via un bloc de communication bidirectionnelle configurable associé BCOM dédié à ces deux unités ;
- l'unité UT d'entrée de chaque bloc BE, qui correspond à la première unité UT sur la ligne correspondant au bloc, est apte à recevoir en entrée des données reçues sur les entrées E1 ou E2 du bloc, et l'unité UT de sortie du bloc, qui correspond à la dernière unité UT sur la ligne correspondant au bloc, est apte à

transmettre en sortie des données traitées à la sortie S du bloc ;

- le contrôleur CG est apte, pour chaque bloc BE, à recevoir de l'extérieur des instructions pour le traitement de données par chacune des unités UT du bloc et à les mémoriser, à initialiser les configurations de chaque unité UT et de chaque bloc BCOM du bloc BE et à commander leur reconfiguration dynamique.

10 5. Composant selon la revendication 4 dans lequel :

- les blocs BCOM de chaque paire de blocs BCOM, dont les unités UT associées sont, selon le réseau matriciel d'unités du bloc TD, sur des colonnes adjacentes et sur des lignes adjacentes, sont aptes à échanger des données par un registre inter-lignes associé REG, de façon à permettre l'échange de données entre unités UT d'une ligne à l'autre;
- le contrôleur CG est apte à commander la reconfiguration dynamique des unités UT et des blocs BCOM du bloc TD pour rendre le bloc TD apte à traiter des données en mode à lignes dépendantes, pour au moins deux lignes adjacentes du réseau matriciel, en commandant la bifurcation des données entre deux blocs BE correspondant aux deux lignes adjacentes, via un registre inter-lignes REG entre ces deux lignes.

30 6. Composant selon la revendication 5 dans lequel le contrôleur CG est apte à transmettre, via un bus de contrôle, des signaux de type RESET ou ENABLE ou d'horloge à chaque registre REG du bloc TD, et dans

lequel chaque registre inter-lignes REG entre deux blocs BCOM est apte à :

- stocker des données reçues d'un bloc BCOM dans une mémoire et les rendre disponibles pour l'autre bloc BCOM ;
- recevoir des signaux de type RESET ou ENABLE ou d'horloge émis par le contrôleur CG et exécuter les commandes correspondantes à ces signaux.

7. Composant selon l'une quelconque des revendications 4 à 6 dans lequel : les entrées E1 et E2 d'un bloc BE sont celles d'un multiplexeur associé au bloc, la sortie du multiplexeur étant connectée à l'entrée de l'unité UT d'entrée du bloc BE pour la transmission de données à cette unité.

8. Composant selon l'une quelconque des revendications 4 à 7 dans lequel : les unités de traitement de données UT du bloc TD sont des bancs de cellules élémentaires synchrones de traitement de données CE configurables, chaque banc comportant au moins une cellule élémentaire CE ; les cellules CE de toute unité UT d'un bloc BE quelconque sont connectées à un bus de contrôle commun qui les relie au contrôleur CG, le contrôleur CG étant apte à contrôler les cellules CE et à commander leur reconfiguration; pour chaque paire d'unités UT adjacentes dans un bloc BE, les cellules élémentaires d'une unité UT sont aptes à échanger des données avec les cellules élémentaires de l'autre unité UT via le bloc BCOM associé à ces unités, le bloc BCOM étant apte à réaliser ces échanges de

données ; chaque cellule CE de l'unité UT d'entrée de
chaque bloc BE, qui correspond à la première unité UT
sur la ligne correspondant au bloc, est apte à recevoir
en entrée des données reçues en entrée de l'unité UT
5 par un bus de données commun à toute cellule CE de
l'unité UT; chaque cellule CE de l'unité UT de sortie
de chaque bloc BE, qui correspond à la dernière unité
UT sur la ligne correspondant au bloc, est apte à
transmettre en sortie des données à la sortie S du bloc
10 par un bus de données commun à toute cellule CE de
l'unité UT.

9. Composant selon la revendication 8 dans
lequel chaque bloc BCOM, associé à des unités UT,
15 comporte :
- un bloc de transmission BT apte à réaliser des
échanges de données entre les cellules élémentaires CE
d'une unité UT et les cellules CE de l'autre unité UT
auxquelles BCOM est associé ; le bloc BT étant apte à
20 être configuré par un contrôleur local de bloc CLB;
- un contrôleur local de bloc CLB apte à recevoir et
interpréter des signaux de contrôle provenant du
contrôleur CG et à configurer le bloc BT, selon des
données de configurations reçues et stockées dans une
25 mémoire de configuration locale MB, pour la
transmission de données ; le contrôleur CLB étant apte
à gérer le chargement de la mémoire MB en fonction de
signaux de contrôle reçus du contrôleur CG ;
- un contrôleur général CG apte à contrôler les
30 contrôleurs locaux CLB du bloc TD.

10. Composant selon l'une quelconque des revendications 8 ou 9 dans lequel chaque cellule élémentaire synchrone CE d'un banc comporte :

- 5 - un bloc opérateur BO apte à recevoir des données en entrée par un bus de données d'entrée, à les stocker et à effectuer un traitement sur ces données selon des signaux de configuration provenant d'un contrôleur local de cellule CLC ; à transmettre le résultat du traitement en sortie par un bus de données de sortie;
- 10 - un contrôleur local de cellule CLC apte à :
 - recevoir des données par le bus de données d'entrée ;
 - gérer les différents modes de fonctionnement de la cellule CE, c'est-à-dire un mode
 - 15 d'initialisation, pour le chargement des mots de configuration de la cellule dans une mémoire de configuration locale MC, un mode de test, pour effectuer des tests structurels de la cellule, et un mode normal pour l'exécution des opérations de
 - 20 traitement de données, en fonction de signaux de configuration émis par le contrôleur CG et reçus par le contrôleur local via le bus de contrôle commun ;
 - gérer le chargement de la mémoire MC en fonction
 - 25 d'un mode de fonctionnement de la cellule indiqué par le contrôleur CG ;
 - interpréter des signaux de configuration reçus du contrôleur CG pour soit charger un mot de configuration correspondant dans un registre de
 - 30 configuration local RCL pour le maintenir pendant le cycle de traitement de données par la cellule,

soit configurer le bloc BO en fonction d'un mot de configuration figurant dans le registre RCL ;

- effectuer des tests structurels de la cellule CE sur réception d'un signal de mode test émis par le contrôleur CG ;

- un contrôleur général CG apte à contrôler les contrôleurs locaux CLC du bloc TD.

11. Composant selon la revendication 10 dans lequel le contrôleur local de cellule CLC est apte à interpréter des signaux de configuration reçus du contrôleur CG pour d'abord lire un numéro de configuration de cellule sur le bus de contrôle commun puis charger un mot de configuration correspondant localement à ce numéro, selon une table de correspondance locale, dans le registre RCL ; les numéros de configuration de cellule étant stockés dans une mémoire du contrôleur CG et le contrôleur CLC étant apte à mémoriser la table de correspondance locale.

20

12. Composant selon l'une quelconque des revendications 10 ou 11 dans lequel le contrôleur CG est apte à recevoir d'une cellule CE une indication de dysfonctionnement résultant d'un test structurel mis en œuvre sur cette cellule, et à commander au contrôleur local CLC de la cellule CE de configurer la cellule en mode BYPASS.

13. Composant selon les revendications 9 et 12 dans lequel le contrôleur local de bloc CLB d'un bloc de communication BCOM, connecté à une unité UT,

est apte à configurer le bloc BCOM en mode BYPASS lorsque le contrôleur CG, ayant reçu d'une ou plusieurs cellules CE de l'unité UT une indication de dysfonctionnement, transmet au contrôleur CLB un signal de passage en mode BYPASS ; le contrôleur CG étant apte à commander le passage en mode BYPASS d'un bloc BCOM d'une unité UT lorsqu'une commande de configuration en mode BYPASS d'une cellule CE de l'unité UT a été transmise au contrôleur local CLC de la cellule CE.

10

14. Composant selon l'une quelconque des revendications 8 à 13 dans lequel le résultat du traitement de données effectué par une cellule CE est apte à être stocké dans un registre local ; un résultat stocké dans ce registre local étant maintenu lors d'une reconfiguration de la cellule.

15

15. Composant selon l'une quelconque des revendications 1 à 14 comportant :

- 20 - un bloc de gestion d'entrée GE apte à recevoir des données de l'extérieur et à stocker temporairement ces données reçues, à formater des données stockées selon le mode d'initialisation ou le mode normal indiqué par le contrôleur CG via un bus de contrôle propre, à
- 25 transmettre des données formatées à l'entrée E1 des blocs BE du bloc TD via le bus de données d'entrée commun ;
- un bloc de gestion de sortie GS, connecté au contrôleur CG par un bus de contrôle propre pour la
- 30 commande de réception de données traitées, connecté au bus de données de sortie de chacun des blocs BE du bloc

TD pour recevoir des données traitées, et apte à reformater des données traitées reçues, à stoker dans une mémoire tampon des données reformatées, à recevoir une requête de transmission depuis l'extérieur, à
5 transmettre à l'extérieur des données stockées dans la mémoire tampon sur requête reçue de l'extérieur.

16. Composant selon l'une quelconque des revendications 10 à 15 dans lequel le contrôleur local
10 CLC de cellule CE, le contrôleur local CLB de bloc BCOM et le contrôleur général CG sont aptes à mettre en œuvre une méthode JTAG pour le chargement des configurations en phase de reconfiguration dynamique du composant, et pour la réalisation de tests ; le
15 contrôleur CG comprenant un contrôleur TAP connecté par des bus série JTAG, bloc BE par bloc BE, aux cellules CE et aux blocs BCOM du bloc TD.

17. Composant selon l'une quelconque des revendications 4 à 16 dans lequel, sur les lignes du
20 réseau, les unités UT sont en nombre pair N et les blocs BCOM sont répartis dans N/2 cellules de base CB, chaque cellule CB comportant deux unités UT adjacentes et le bloc BCOM associé à ces unités ; le contrôleur CG
25 étant apte à commander la reconfiguration dynamique des cellules CB du réseau et des blocs BCOM de communication entre cellules CB, ligne par ligne, pour rendre le bloc TD apte à traiter des données au moyen des cellules CB.

18. Composant selon la revendication 17,
pour le traitement d'un ensemble d'applications
comportant au moins une application, dans lequel chaque
cellule de base CB du bloc TD configurée pour effectuer
5 un traitement est apte à réaliser les opérations
nécessaires au traitement complet d'au moins une
application de l'ensemble, chaque application de
l'ensemble étant apte à être réalisée complètement par
au moins une cellule de base CB.

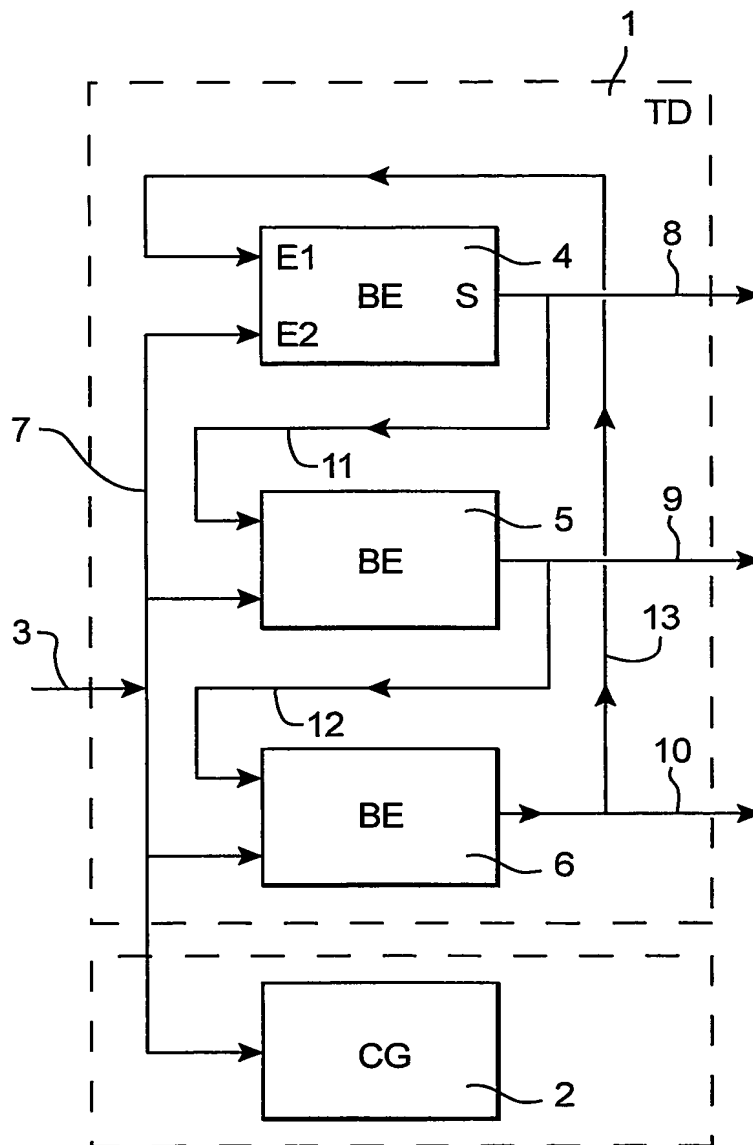


FIG. 1

2 / 11

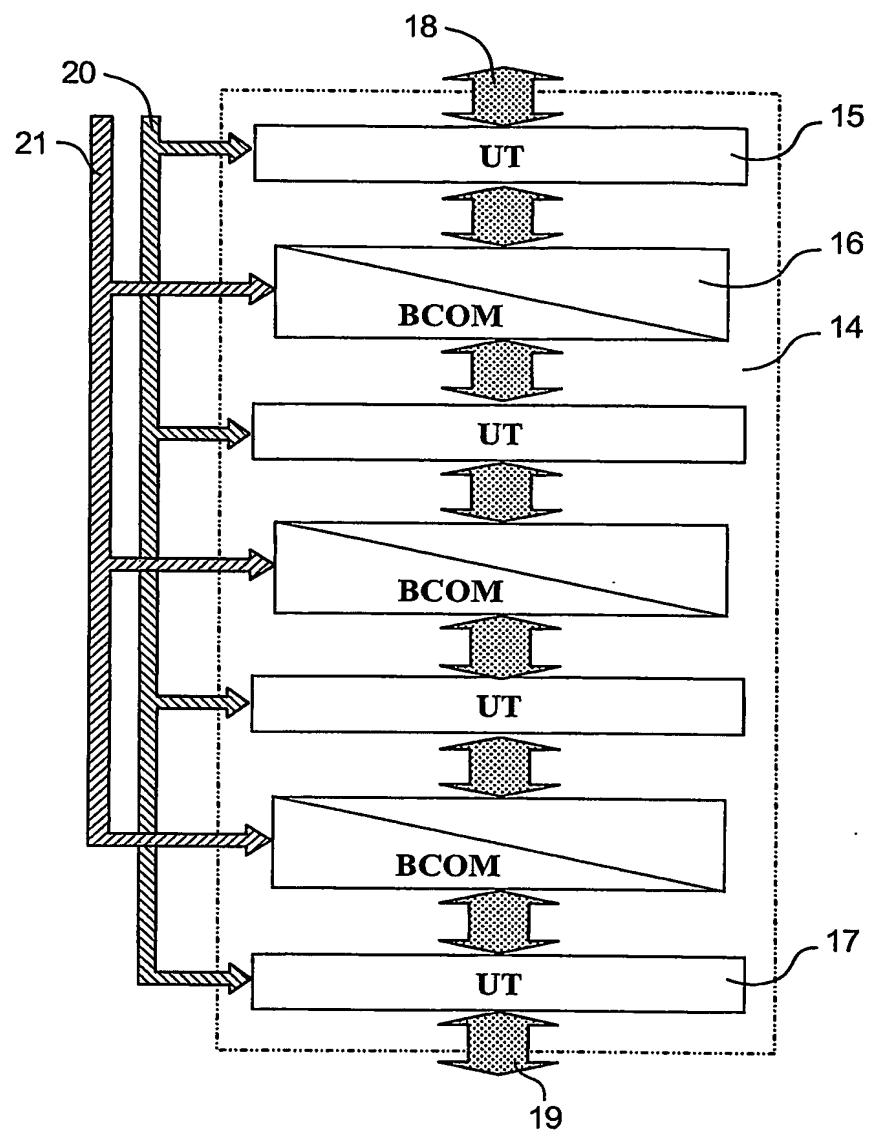


FIG. 2

3 / 11

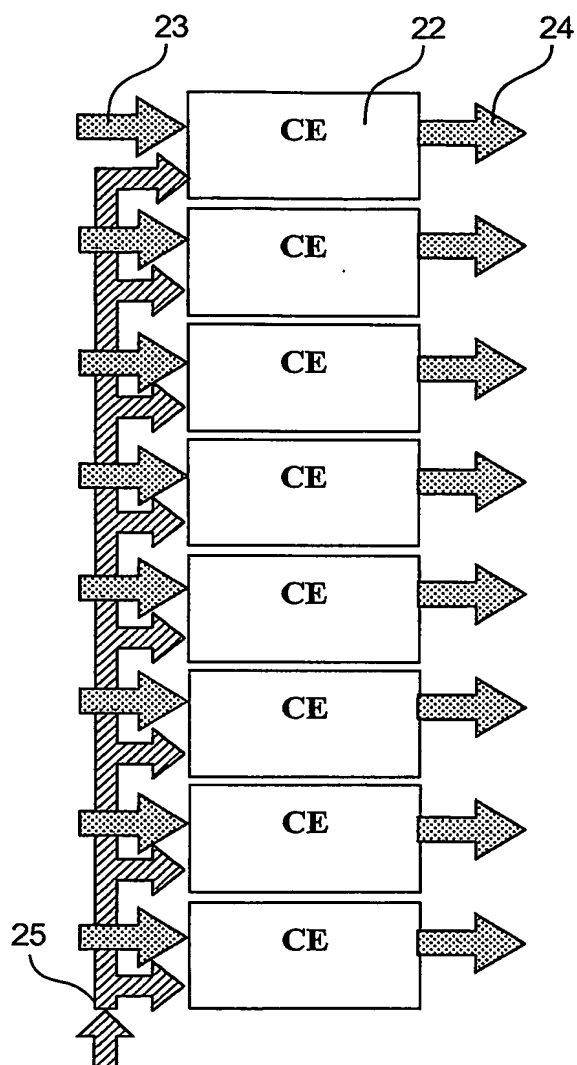


FIG. 3

4 / 11

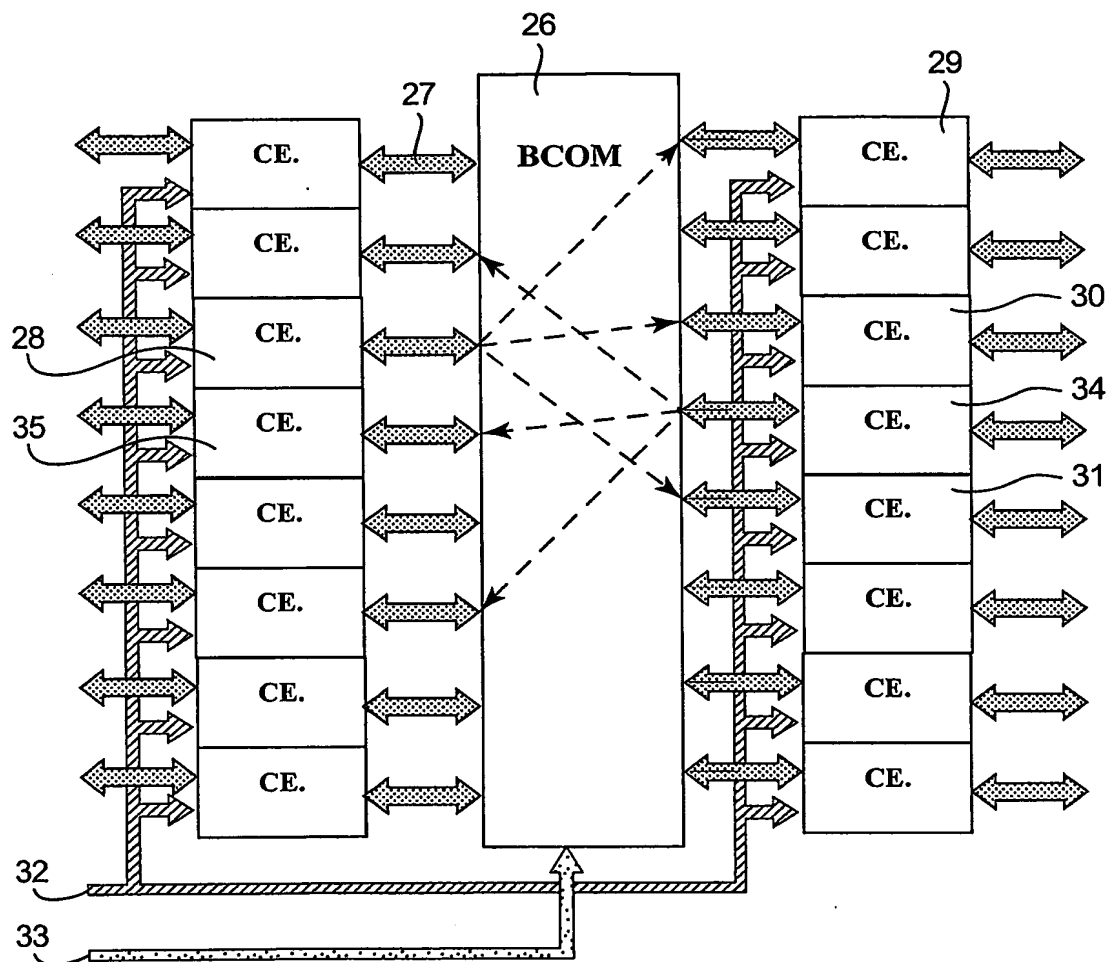


FIG. 4

5 / 11

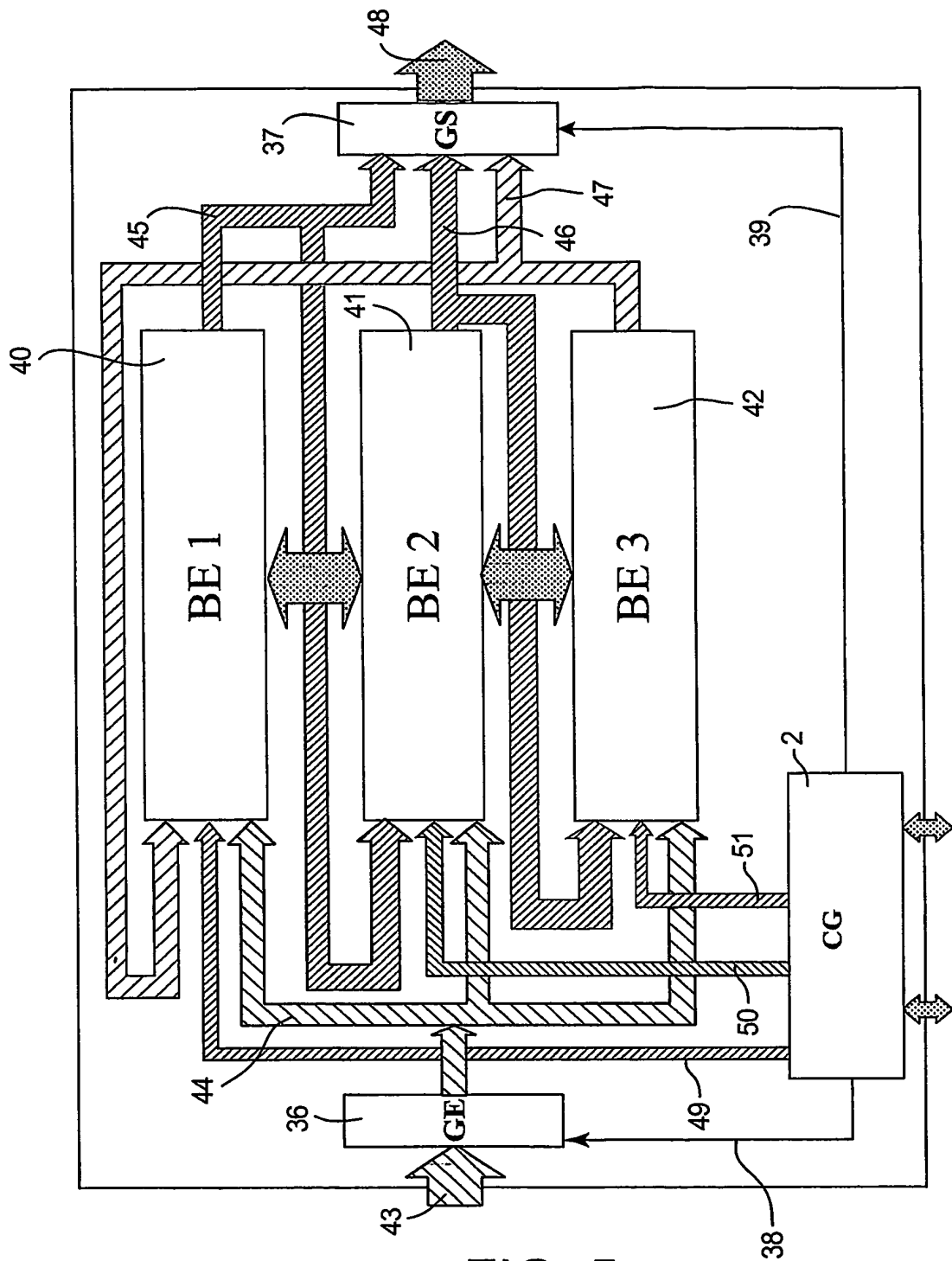


FIG. 5

6 / 11

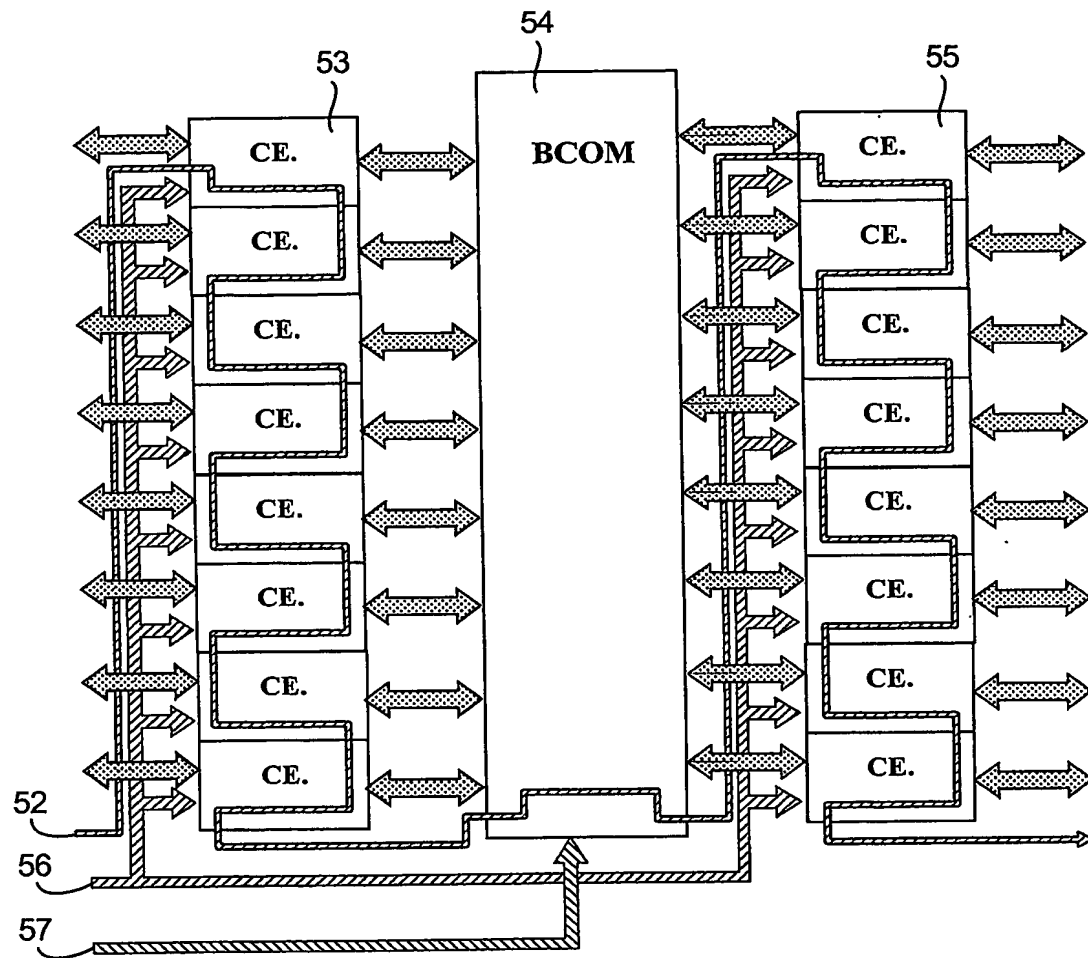


FIG. 6

7 / 11

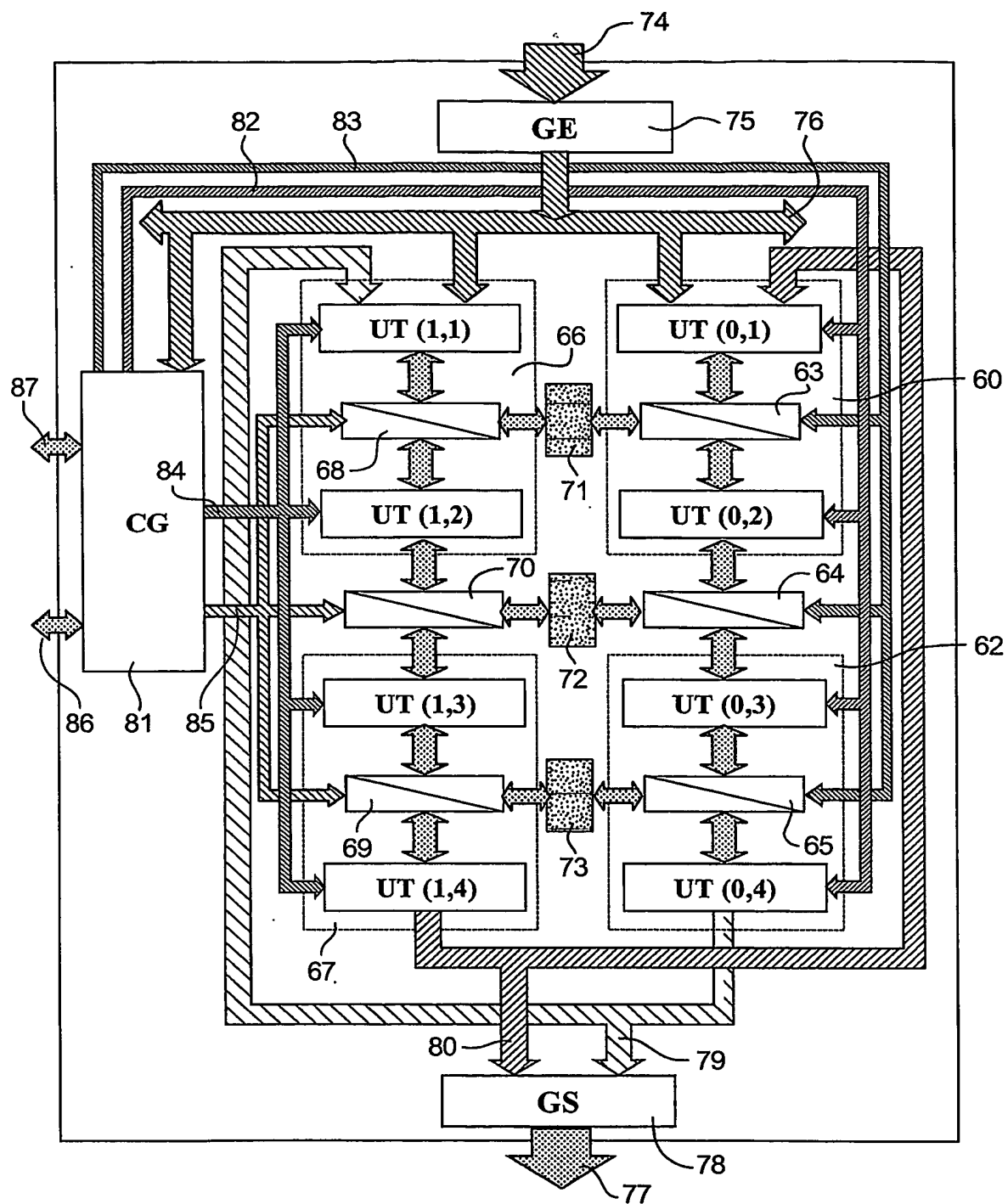


FIG. 7

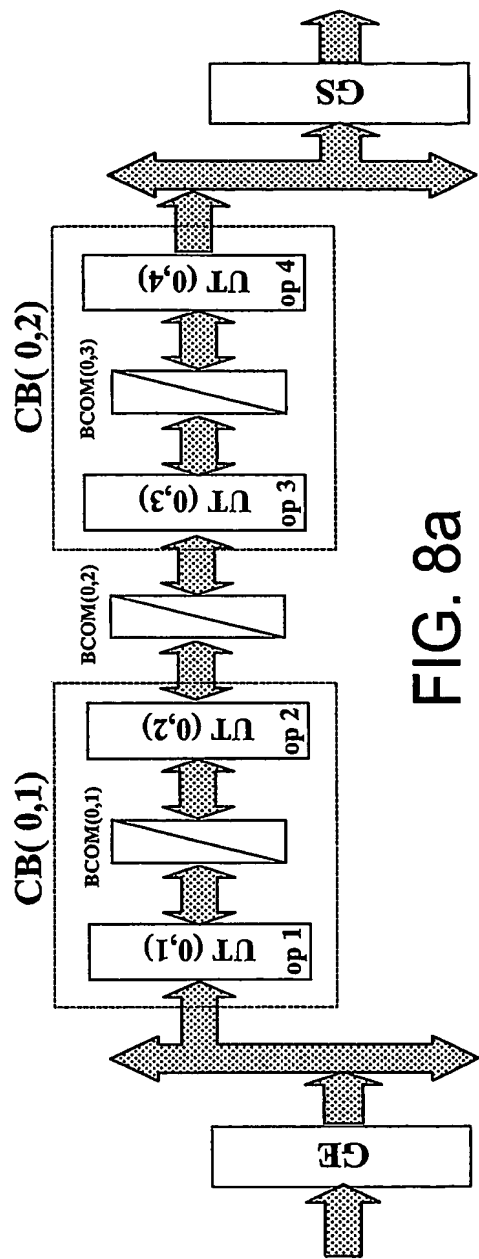


FIG. 8a

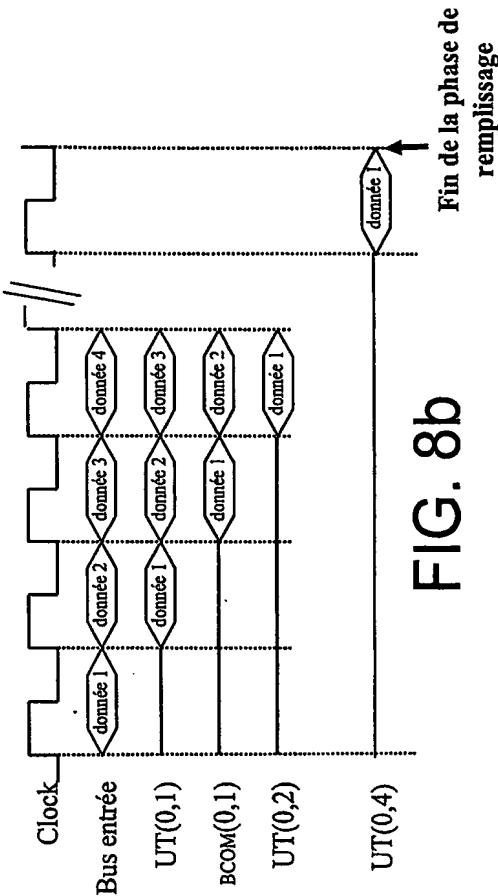


FIG. 8b

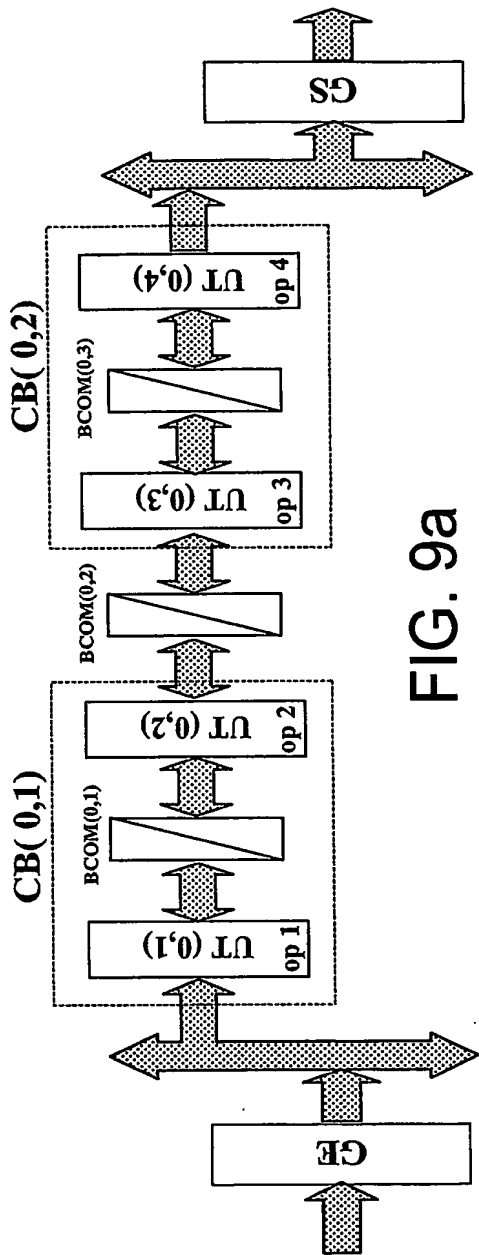


FIG. 9a

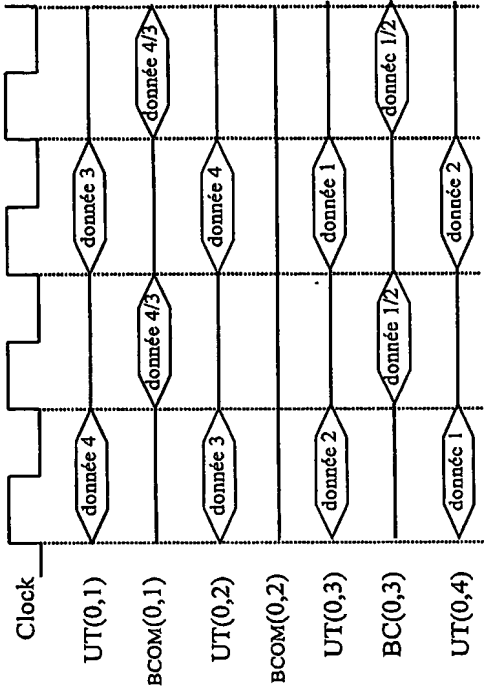


FIG. 9b

